CLAIMS

Sub A1

1. A method of generating a file suitable for programming a programmable logic device, the method comprising the steps of:

(A) generating a programming item from a plurality of parameters that define a program for said programmable logic device;

(B) compressing said programming item to present a compressed item;

(C) storing said programming item in a programming field of said file in response to generating; and

(D) storing said compressed item in a non-programming field of said file in response to compressing.

2. The method according to claim 1, further comprising the step of storing at least one of said parameters in a second non-programming field of said file.

3. The method according to claim 1, further comprising the step of generating a dictionary for compressing prior to compressing said programming item.

4. The method according to claim 3, wherein said dictionary is generated independently of said compressing step.

5. The method according to claim 4, wherein said compressing is a Huffman encoding and said dictionary is a Huffman tree.

6. The method according to claim 1, further comprising the step of encoding said compressed item from a binary representation to a symbol representation in response to compressing.

7. The method according to claim 6, further comprising the step of mapping said symbol representation to a character representation in response to encoding.

8. The method according to claim 1, further comprising the steps of:

generating an error detection item; and

storing said error detection item in a second non-programming field of said file.

9. The method according to claim 8, further comprising the steps of:

extracting said error detection item from said file;

extracting said compressed item from said file;

decompressing said compressed item to present a backup programming item; and

validating said backup programming item with said error detection item.

10. The method according to claim 1, wherein said steps (A) through (D) are stored in a storage medium as a computer program that is readable and executable by a computer to generate said file.

Claims 11-20 OK for 714/725 DE

11. A method of repairing a file suitable for programming a programmable logic device, the method comprising the steps of:

(A) extracting an error detection item from a non-programmable field of said file;

(B) extracting a compressed item from a second non-programmable field of said file;

(C) decompressing said compressed item to produce a backup programming item; and

(D) validating said backup programming item with said error detection item.

12. The method according to claim 11, further comprising the steps of:

extracting a programming item from a programmable field of said file; and

validating said programming item with said error detection item.

13. The method according to claim 12, further comprising the step of replacing said programming item with said backup programming item in response to validating said backup programming item.

14. The method according to claim 11, further comprising the step of generating a dictionary for decompressing prior to decompressing said compressed item.

15. The method according to claim 14, wherein said dictionary is generated independently of said decompressing step.

16. The method according to claim 15, wherein said decompressing is a Huffman decoding and said dictionary is a Huffman tree.

17. The method according to claim 11, further comprising the step of mapping said compressed item from a character representation to a symbol representation in response to extracting said compressed item.

18. The method according to claim 17, further comprising the step of decoding said compressed item from said symbol representation to a binary representation in response to mapping.

19. The method according to claim 11, wherein said steps (A) through (D) are stored in a storage medium as a computer program that is readable and executable by a computer to repair said file.

20. An apparatus comprising:

means for extracting an error detection item from a non-programmable field of a file suitable for programming a programmable logic device;

means for extracting a compressed item from a second non-programmable field of said file;

means for decompressing said compressed item to produce a backup programming item; and

means for validating said backup programming item with said error detection item.